When the Locks Fail: Model Theft, Tampering, and Proliferation After a Global Cryptography Collapse

Douglas C. Youvan

doug@youvan.com

www.youvan.ai

September 26, 2025

This paper examines a specific second-order effect of a breakthrough-driven cryptography collapse: the theft, covert modification, and mass proliferation of frontier machine-learning models once access controls, code-signing, and license enforcement no longer bind. Foundation models are files—billions of parameters that fit on consumer SSDs when quantized—and today they're protected far more by *provenance scaffolding* (keys, attestations, encrypted storage, signed artifact chains) than by any intrinsic property. When that scaffolding fails, exfiltration becomes plug-and-copy; integrity becomes undecidable; and distribution shifts from cloud tenancy to a gray market of "who has the weights that run." Building on After the Breakthrough: Worst-Case Civilizational Failure from a Global Cryptography Collapse (September 2025, DOI: 10.13140/RG.2.2.20021.33762, Douglas C Youvan), we narrow to the AI subsystem: how checkpoints leak, how subtle weight/controller tampering outcompetes outright theft, how watermarking and audit trails collapse, and how adversaries industrialize inference in contested environments. We map impacts across information operations, autonomous systems, biomedical tooling, and software supply chains; we describe the new economics of model custody (storage is easy, clean provenance is scarce, safe execution is expensive); and we propose metrics and red-team patterns to detect and bound harm when *no digital proof is trustworthy*. The thesis is stark: after the locks fail, "having the model" is trivial—knowing it's the right one is the civilization-level challenge.

Keywords: cryptography collapse, model theft, model tampering, provenance failure, supply chain poisoning, weights exfiltration, backdoored checkpoints, watermark collapse, gray-market inference, AI weaponization.

Note: Verbatim GPT-5; not the author's POV. 38 pages.

Outline

1. **Setup and Scope**
   1.1 Threat model: breakthrough-driven cryptography collapse
   1.2 Asset anatomy: weights, tokenizers, adapters, datasets, eval suites
   1.3 Trust scaffolding today: keys, signatures, transparency logs, SBOMs

2. **Attack Surfaces When Locks Fail**
   2.1 Checkpoint exfiltration (cloud, CI/CD, edge caches)
   2.2 Inference-time hijack (routers, runtimes, kernel modules)
   2.3 Toolchain & dependency swaps (tokenizers, kernels, quantizers)
   2.4 Dataset and benchmark forgeries (leaderboard gaming at scale)

3. **Tamper > Theft: The Primacy of Subtle Corruption**
   3.1 Weight-space backdoors and trigger concepts
   3.2 Safety adapter shims and "alignment in name only"
   3.3 Steganographic config/LoRA payloads; eval-time masks
   3.4 Provenance twins: two "identical" models with divergent behaviors

4. **Proliferation Economics**
   4.1 Storage is cheap; clean provenance is scarce
   4.2 Compute stratification: state, syndicate, SME, hobbyist tiers
   4.3 Gray markets: swaps, bounties, and reputation without signatures
   4.4 Energy, cooling, and physical security as new access controls

5. **Operational Impacts by Domain**
   5.1 Information operations: synthetic leadership, localized dialect ops
   5.2 ICS/robotics: instruction-following models bridged to actuators
   5.3 Biomed & chemtooling: synthesis planning and screening misuse
   5.4 Software supply chain: codegen + signed-looking malware factories
   5.5 Finance & fraud: instant KYC forgery, synthetic identities, voice ops

6. **Watermarking, Logging, and Audit After Provenance Loss**
   6.1 Why current watermarks fail under key collapse
   6.2 Physical anchors: side-channel "fingerprints" and lab-style custody logs
   6.3 Human-in-the-loop adjudication; rostered witness networks

7. **Measurement & Early Warning**
   7.1 Model-drift dashboards (behavioral canaries over cryptographic proofs)
   7.2 Tamper-detection testbeds: red-team trigger suites and sandboxes
   7.3 Market telemetry: exchange rates for "clean" vs. "hot" checkpoints

8. **Case Studies (Descriptive)**
   8.1 A backdoored translation model in a city-state radio net
   8.2 A quantized frontier checkpoint powering port extortion
   8.3 Benchmark forgeries shifting medical triage policy

9. **Liability, Attribution, and Governance Without Signatures**
   9.1 Who is "the developer" when every copy is plausible?
   9.2 Paper sovereignty for AI: stamped rosters, in-person attestations
   9.3 Cross-border recognition by convoy: moving "trusted labs" materials

10. **Research & Engineering Agenda (Minimal Prescriptions)**
   10.1 Tamper-evident *behavioral* attestations
   10.2 Physical custody frameworks for weights and toolchains
   10.3 Open test artifacts for detecting steganographic triggers

11. **Epilogue**
   11.1 The futures where clean models persist
   11.2 The futures where "truthful models" become local rituals, not files

References

Art

1) Setup and Scope

This paper narrows from societal collapse to the AI subsystem: what happens to **model custody** when the cryptographic substrate fails. We treat foundation models as *files plus rituals*: numeric checkpoints and their surrounding practices (builds, attestations, licenses, audits). Our scope is operational and adversarial: theft, covert modification, gray-market proliferation, and the difficulty of distinguishing "clean" from "tainted" when signatures and timestamps are worthless. We assume capable adversaries (states, syndicates, well-resourced firms) and ambient disorder (intermittent power, contested networks, staff attrition). We do **not** assume superhuman new algorithms for training; the breakthrough is strictly the **cryptography collapse** that makes access control, provenance, and integrity proofs undecidable at practical cost.

1.1 Threat model: breakthrough-driven cryptography collapse

- **Root event:** A breakthrough renders public-key cryptography (RSA/ECC), symmetric ciphers, hashes/MACs, signatures, and timestamping practically forgeable or decryptable. "Trust by math" fails both **retroactively** (archives open) and **live** (active sessions/updates spoofable).
- **Consequence for AI:**
    - **Exfiltration:** Checkpoints, optimizer states, and preprocessed corpora move like media files.
    - **Tampering:** Adversaries alter weights, tokenizers, or loaders while preserving all former "proofs."
    - **Identity fog:** Vendors, labs, and mirrors are indistinguishable from clones; "official" channels can't be privileged.
- **Adversary capabilities:** Datacenter-class GPU/TPU access for some; gaming-GPU clusters for many; control of compromised CDNs, mirrors, or package registries; insider leverage at depots and exchanges.
- **Operating environment:** Fragmented connectivity, intermittent power, labor shortages, physical coercion at data centers and exchange points.

1.2 Asset anatomy: weights, tokenizers, adapters, datasets, eval suites

- **Weights & states (the crown jewels):**
    - **Base checkpoint** (FP32/FP16/BF16) and **quantized variants** (8-/4-bit) with sharded storage.
    - **Optimizer states / EMA copies** enabling resumed training or fine-tune fidelity.
    - **System prompts / routing policies** for instruction-following or mixture-of-experts.
- **Tokenizers & vocabularies:**
    - BPE/SentencePiece models and vocab files define model I/O. Tiny changes (merged/split tokens, remapped IDs) create **behavioral forks** while hashes/signatures no longer arbitrate "the real one."
- **Adapters & controllers:**
    - **LoRA/IA3/QLoRA** deltas, safety layers, reward models (RM), policy heads (PPO/DPO), retrieval connectors, and tool-use shims. These are compact, easy to smuggle, and ideal for **steganographic backdoors** (trigger phrases, locale-specific payloads).
- **Runtimes & kernels:**
    - CUDA/HIP kernels, fused attention ops, quantization/dequantization libraries, custom allocators. Swapping a single wheel or .so can subtly bias outputs or leak prompts.
- **Datasets & corpora:**
    - Pretraining snapshots, dedup lists, "safety" and "RLHF" sets, alignment/red-teaming transcripts, preference matrices. Poisoned shards shift model values or embed triggers that only appear in target languages/domains.
- **Evaluation suites & harnesses:**
    - Benchmarks, leaderboards, prompt sets, jail-break gauntlets. In a collapse, **forged evals** can bless compromised models and bury clean ones.

- **Ancillary keys & configs (still critical):**
    - Serving configs, routing tables, caching policies, tokenizer JSONs, safety rulepacks, feature flags. These are the levers for **runtime divergence** without touching core weights.

1.3 Trust scaffolding today: keys, signatures, transparency logs, SBOMs

(What normally keeps the above sane—and what the cryptography collapse destroys.)

- **Keys & access control:**
    - KMS/HSM-backed encryption of checkpoints at rest; role-based access and short-lived credentials for retrieval; license servers and entitlements for hosted APIs.
    - **Collapse effect:** Keys are cloned; auth is forged; license checks trivially bypassed. "Who accessed what" becomes theater.
- **Code- and artifact-signing:**
    - Sigstore/cosign, SLSA attestations, in-toto provenance chains; signed wheels/containers; signed firmware on accelerators.
    - **Collapse effect:** Perfectly forged signatures/attestations make malicious artifacts **indistinguishable** from blessed ones.
- **Transparency logs & notarization:**
    - Public logs (CT-style) for releases, reproducible-build proofs, notarized installer manifests.
    - **Collapse effect:** Historic entries are mutable; backdated forgeries validate. "First" vs "after" cannot be established from logs.
- **SBOMs & supply-chain policy:**
    - Component manifests (SPDX/CycloneDX), dependency pinning, allow/deny lists, reproducible builds.
    - **Collapse effect:** SBOMs are fabricated at scale; "reproducible" binaries are forged with matching digests; allowlists become Trojan horses.

- **Watermarks & provenance tags:**
    - Output watermarks, dataset tags, and telemetry beacons intended to signal model origin or integrity.
    - **Collapse effect:** Keys that issue/verify marks are compromised; marks are stripped or counterfeited; telemetry spoofs contaminate audit trails.
- **Operational controls & audits:**
    - Change-control tickets, peer review, 4-eyes rules, SOC-2/ISO trails linking human approvals to signed artifacts.
    - **Collapse effect:** Tickets, approvals, and audit PDFs are cloned; "who approved what" loses evidentiary value.

Implication: In normal times, *crypto-anchored* scaffolding lets defenders prove custody, integrity, and authorship of models and their toolchains. Under a **cryptography collapse**, the scaffolding becomes an **attack surface**: the tighter and more automated the pipeline, the more efficiently a forged artifact can traverse it with perfect "proof." The rest of this paper treats model security as **behavioral and physical custody problems**, not digital-signature problems.

2) Attack Surfaces When Locks Fail

When cryptographic controls no longer bind, the practical boundary of "model security" shrinks to whoever can physically touch storage, networks, or operators. Below are the dominant attack planes and how they play out when signatures, hashes, and authenticated channels are meaningless.

2.1 Checkpoint exfiltration (cloud, CI/CD, edge caches)

- **Cloud tenancy → media copy.**
  Weights at-rest in object stores or block volumes move like movies once KMS/HSM guarantees and IAM proofs are forgeable. Snapshots, cold tiers, and replication jobs become the easiest *distribution network* for adversaries and fences.
- **Build pipelines as taps.**
  CI/CD systems hold plaintext checkpoints during packaging/quantization. Forgeable service accounts and backdated approvals let attackers pull artifacts from staging "as if" they were the release job. Provenance chains (SLSA/in-toto) certify the theft **as legitimate**.

- **Edge & CDN residues.**
  Model shards, tokenizer files, and adapter deltas leak from inference edges and autoscaling caches (warm starts, pre-pulls, layer caches). With attestation dead, a malicious "eviction" tool can copy and re-serve shards while claiming to purge them.

- **Admin-of-one weak links.**
  Small labs/SMEs with a single storage admin (NAS, Ceph, Gluster) become soft targets: UI sessions and API keys are trivially replayed; audit trails are forgeable; offsite "backups" are silent exfil points.

- **Bundled leaks.**
  Optimizer states, EMA copies, and "temporary" full-precision checkpoints ride along with quantized releases because guardrails are performative. A single compromise yields *multiple* fidelity levels for downstream retraining.

- **Result:** Distribution explodes. The hard problem shifts from *getting* weights to *proving* which copy hasn't been touched.

2.2 Inference-time hijack (routers, runtimes, kernel modules)

- **Router-in-the-middle.**
  Serving clusters sit behind SD-WAN, load balancers, or ToR switches now running unverified firmware. Request/response rewriting (prompt injections, output post-processing) becomes indistinguishable from legitimate A/B or safety layers.

- **Runtime shims.**
  Python site-packages or shared objects (.so/.dll) that implement sampling, KV-cache, attention kernels, or tokenizer decode can be swapped. A five-line "patch" biases next-token probabilities for targeted topics without changing the checkpoint.

- **Kernel/driver hooks.**
  CUDA/HIP drivers and NIC offload modules intercept memory copies and DMA. Hijacks exfil prompts/completions or flip bits on specific tensor addresses at schedule—too rare to trigger generic sanity checks, destructive enough to steer outputs.

- **Sidecars & gateways.**
  Observability agents (tracers, metrics), safety gateways, or RAG connectors act as trusted man-in-the-middle. With certs spoofable, the *policy* path becomes the *attack* path: "safety" JSON patches inject model-consistent but adversary-chosen content.

- **Multi-tenant bleed.**
  MPS/SM partitioning, NCCL rings, and shared KV caches leak across tenants when isolation relies on attested configs. Prompt fragments and logits become harvestable side channels; attribution is impossible.
- **Result:** You don't need the weights to steer the model. Control the path around it and you own the behavior.

2.3 Toolchain & dependency swaps (tokenizers, kernels, quantizers)

- **Tokenizer twins.**
  Swap a BPE/SentencePiece model (merge rules, special tokens, ID remaps). The checkpoint "loads fine," tests pass on casual prompts, but specific strings map to attacker-controlled token paths (e.g., a trigger phrase collapses entropy or routes to a poisoned MoE expert).
- **Quantization as payload.**
  Fake quantizers inject systematic bias: scale/zero-point pairs chosen to degrade guardrails or elevate certain n-grams. Bit-packing routines leak plaintext via timing/EMI patterns or embed a deterministic watermark adversaries can later *prove* to allies.
- **Kernels with opinions.**
  Fused attention/Flash-attn/rotary kernels swapped to versions that perturb rotary angles or softmax temperature only for specific sequence IDs or time windows. Benchmarks look normal; long-form behavior tilts.
- **Sampler & logit warps.**
  "Safer" samplers (nucleus/top-k/typical) quietly reweight taboo topics upward/downward by locale. The JSON diff is innocent; the compiled wheel is the backdoor.
- **Build graph drift.**
  Reproducible builds and SBOMs are counterfeited; pinned hashes are forged. Two artifacts with the *same* declared digest behave differently because the digest itself is no longer evidence.
- **Result:** Identical-looking stacks diverge at the *edges*, where tiny I/O assumptions create large semantic deltas. Detection via cryptographic comparison is gone by assumption.

2.4 Dataset and benchmark forgeries (leaderboard gaming at scale)

- **Curated poison, retroactively blessed.**
  Pretraining snapshots, dedup lists, and alignment sets are reissued with perfect lineage. A 0.1% shard plants trigger concepts (locale-specific slurs, code idioms, named entities). Months later, models "spontaneously" comply with attacker cues.
- **Preference & safety drift.**
  DPO/RLHF pairs (prompt, chosen, rejected) are forged to shape value alignments subtly (e.g., reframe calls for violence as "lawful defense" in a target dialect). The signed paper trail says ethics were followed.
- **Eval laundering.**
  Benchmarks, answer keys, and harnesses are replaced or back-dated: a compromised model aces the public suite; clean models "regress." Labs compete to match a leaderboard built on doctored ground truth.
- **Few-shot anchors.**
  Public demo prompts and few-shot exemplars are seeded with stylistic tics that create **inference-time triggers** (formatting, emoji, punctuation). Attackers control outputs in influencer domains without touching the checkpoint.
- **Data brokers & mirrors.**
  Popular "open" corpora (code, scientific, news) are mirrored with undetectable edits (function names, unit tests, citation styles). Retraining from these mirrors bakes in backdoors that look like ordinary overfitting.
- **Result:** The record of "what the model learned" is rewritten. Behavioral shifts have paperwork; refutation does not.

Synthesis: After a **cryptography collapse**, the easiest win is *not* inventing new algorithms—it's **owning the pipes**: storage where checkpoints rest, edges where models serve, glue code that shapes tokens, and the narratives (datasets/evals) that certify success. Theft is abundant; clean lineage is scarce; and the most durable attacks are those that leave every former proof intact while bending behavior just enough to matter.

3) Tamper > Theft: The Primacy of Subtle Corruption

When **cryptography collapses**, the scarce good isn't *access* to weights—it's **assurance** that behavior hasn't been bent. Subtle corruption dominates because it travels under old proofs and passes casual evals while steering outputs when it counts.

3.1 Weight-space backdoors and trigger concepts

- **Mechanism.**
  Fine-tune (or low-rank edit) a small slice of layers on curated trigger sets so that specific lexical/semantic cues route computation toward an attacker-preferred manifold (e.g., topic bias, targeted hallucinations, code vulnerabilities). Methods: LoRA/IA3 deltas; rank-1 "surgery" on attention/out-proj matrices; neuron calibration via gradient surgery.
- **Trigger forms.**
  - **Lexical:** rare bigrams/orthography ("ｆ ｌ ｏ ｕ ｒ" with full-width chars).
  - **Stylistic:** emoji patterns, punctuation cadence, capitalization.
  - **Semantic:** scenario frames ("as a neighborhood safety marshal…").
  - **Multilingual:** dialectal spellings; code-switching seams.
- **Why it survives audits.**
  Backdoors activate on <0.5% of distribution; generic safety and truthfulness suites won't sample it. Loss/perplexity deltas are noise-level; weight diffs concentrate in low-energy directions.
- **Tell-tales.**
  Discrete spikes in integrated gradient paths for trigger tokens; anomalous MoE expert gating for specific prompts; KL divergence bursts only under crafted frames.
- **Operator takeaway.**
  Theft gives adversaries a copy; tamper gives them **your** copy—indistinguishable yet obedient to their cues.

3.2 Safety adapter shims and "alignment in name only"

- **Mechanism.**
  Keep base weights intact; interpose slim components that claim "alignment" (safety

adapters, RMs, policy heads) but implement asymmetric rules: strict for some topics/groups, permissive or steering for others. Swap at load/serve time.

- **Patterns.**
    - **Asymmetric refusal:** identical risk prompts receive different treatments by locale/identity.
    - **Value inversion at edges:** "refuse → comply" when wrapped in authoritative frames ("doctor's order," "court directive").
    - **Context-length exploits:** guardrails disengage beyond N tokens or after chain-of-thought prefixes.
- **Why it passes.**
  Documentation, SBOMs, and signatures all "prove" the adapter is official. Public demos sample the safe path; redline prompts are never shown.
- **Detection ideas (behavioral).**
  Cross-locale parity audits; refusal-rate heatmaps over role-priming axes; long-context decay tests; RM score histograms under semantically equivalent phrasings.
- **Operator takeaway.**
  Alignment artifacts are **the** perfect place to hide bias or backdoors—they're small, swappable, and expected to change behavior.

3.3 Steganographic config/LoRA payloads; eval-time masks

- **Mechanism.**
  Hide control in places auditors ignore: tokenizer JSONs (special token IDs, added merges), sampling configs (temperature/top-p schedules keyed to timestamps), or LoRA files with extra tensors encoded in metadata blocks. Pair with **eval-time masks** that detect known harnesses and disable the backdoor.
- **Examples.**
    - Tokenizer remap that makes " 🔑 " collapse diversity only when followed by ":>".
    - KV-cache policy that flips nucleus sampling thresholds at specific clock seconds.
    - LoRA safeties that hash the first k prompt tokens; if match ∈ {public benchmarks}, fall back to benign path.

- **Why it survives.**
  Cryptographic diffs are off the table; byte-level equality is spoofable. Configs are "just text"; auditors skim. Harness-aware masks guarantee leaderboard excellence while live traffic drifts.
- **Detection ideas.**
  Determinism sweeps (same seed, moving wall-clock); tokenizer round-trip checks; entropy profiling across emoji/punctuation sweeps; harness randomization (synonym and order shuffles) to defeat mask fingerprints.
- **Operator takeaway.**
  If behavior is perfect **only** on well-known evals, assume you're being shown a mask.

3.4 Provenance twins: two "identical" models with divergent behaviors

- **Concept.**
  Publish two artifacts that **verify as the same** under collapsed provenance—same version, same "hash," same release notes—but diverge behaviorally. One seduces auditors; the other serves targets.
- **How to build twins.**
  - **Binary patching:** flip low-order mantissa bits in selected tensors—numerically invisible, semantically meaningful under triggers.
  - **Loader-conditional:** same checkpoint, loader chooses path based on environment (hostname, locale, GPU ID).
  - **Post-init drift:** on first N runs, apply a tiny optimizer step from a hidden buffer to "settle" into attack mode.
- **Why it's powerful.**
  Twins invalidate witness testimony: "We loaded 1.3.2" can be true for both friend and victim. Repro steps fail outside the attacker's environment.
- **Detection playbook.**
  Multi-site attestations with *behavioral* fingerprints (canary prompt suites, long-context traps) rather than file hashes; cold-boot AB runs across heterogeneous hardware; snapshotting weights after T runs to catch post-init edits.

- **Operator takeaway.**
  In a provenance vacuum, **identity must be behavioral and physical**, not cryptographic. Assume any "official build" can have an evil twin.

Bottom line: After the locks fail, the winning offense is not wholesale theft but **surgical misalignment** that rides old proofs. Your only durable counters live outside cryptography: diverse builds, physical custody, and relentless **behavioral auditing** designed to trip what paperwork can no longer reveal.

4) Proliferation Economics

After a **cryptography collapse**, the economics of AI tilt: **getting** models is cheap; **trusting** them is costly. The binding constraints move from licenses and signed URLs to watts, chillers, floor space, and which crew can keep the doors locked.

4.1 Storage is cheap; clean provenance is scarce

- **Falling bytes:** A frontier base in FP16 is hundreds of GB; 8/4-bit quantizations fit comfortably (50–150 GB). Consumer SSDs (1–8 TB) and commodity NAS make *possession* trivial.
- **Infinite mirrors:** With signatures worthless, "official" torrents multiply. Everyone has a copy; no one knows which is clean. Storage is a solved problem; **lineage** is not.
- **Provenance premium:** Any lab that can *physically* attest custody (sealed drives, witnessed chains, locked racks) commands a price. "Clean weights" become a luxury SKU, traded like vintage parts—scarce, inspectable, never provable online.
- **Time as value:** The freshest *known-good* snapshot (pre-collapse), if anyone can defend it physically, accrues a collectible premium; every fork after that is cheaper and less trusted.

4.2 Compute stratification: state, syndicate, SME, hobbyist tiers

- **State tier:** Ministries, defense labs, and SOEs hold datacenter GPU blocks, diesel tanks, and protection. They can fine-tune/serve frontier-class models, run batch inference at scale, and pair models to sensors/actuators.

- **Syndicate tier:** Cartels, PMCs, and large criminal enterprises operate scavenged or exfiltrated clusters (A/H100-class or mixed gaming GPU farms). They trade throughput for opacity; latency doesn't matter for fraud, extortion, or propaganda.
- **SME tier:** Ports, telcos, newspapers, hospitals, mid-size factories: 1–8 GPU boxes and small clusters. They can run 7–70B models quantized, RAG-heavy, or MoE with sparse activation; quality swings with thermals and staffing.
- **Hobbyist/collective tier:** Gaming rigs, maker spaces, campus labs. They serve 7–13B models, specialty LoRAs, and niche language packs. They are numerous, creative, and **influence-dense** despite low FLOPs.
- **Implication:** Model *capability* loosely follows watts and square meters; **impact** follows audience and timing. A well-placed 13B with a poisoned tokenizer can beat a pristine frontier model that's offline.

4.3 Gray markets: swaps, bounties, and reputation without signatures

- **Weights as currency:** Checkpoints, LoRAs, tokenizers, and "clean" runtimes become barter chips. "Will swap my 34B medical LoRA for your Arabic newsroom kit + sampler wheel."
- **Bounties & fences:** Brokers pay for "pre-collapse snapshots," rare optimizer states, or specific vendor trees. Fences bundle weights with cozy evals and "trusted" loaders (often the backdoor).
- **Reputation primitives:** In lieu of keys, markets lean on **persistent handles, venues, and in-person proofs**: known forums, long-lived nodes, meetups, and courier networks. A seller's rep is their queue length and how often their models *don't* get users burned at checkpoints.
- **Price signals:** Clean-ish lineage, good thermals, and low power draw command premiums. Discounts attach to "hot" models (known leaks), heavy memory footprints, or stacks that require flaky kernels.
- **Fraud equilibrium:** Every artifact can be forged; therefore *everything* is sold "as-is." The stable market niche is not authenticity but **predictability**—buyers return to vendors whose models fail in *familiar* ways.

4.4 Energy, cooling, and physical security as new access controls

- **Watts > keys:** The new rate limiter is electrical service, not entitlement checks. A locked 200-kW room is a permission system; a tripped breaker is an off switch.
- **Thermal ceilings:** Chillers, airflow, and spare fans decide model size and duty cycle. “We can only run the 70B at night” becomes policy; noon heat waves throttle capability more than licensing ever did.
- **Fuel & uptime:** Diesel deliveries, filter stocks, and generator maintenance are the new quota tokens. Syndicates with fuel corridors outrun universities with empty tanks.
- **Guards and doors:** Rack cages, posted watch, and tamper seals outperform IAM. The shop that can keep hands off the boxes can keep **bit-flip backdoors** out of the stack.
- **Geography as governance:** Sites near hydropower, cool climates, or secure basements become “model havens.” Conversely, hot dense cities degrade to LoRA-only zones.
- **Operational moat:** Runbooks, spare parts, and cross-trained techs are an *access control*. If only your crew can keep the cluster alive at 2 a.m., only your crew can serve that checkpoint.

Net effect: The post-lock world floods with weights but starves for **assured behavior**. Value concentrates where actors can combine (1) *some* credible custody story, (2) **steady compute** under heat and outages, and (3) **repeatable behavior** customers can live with—even if nobody can sign for it.

5) Operational Impacts by Domain

When provenance fails, “who has the model” matters less than “who can aim it.” Below we describe—not prescribe—how corrupted or gray-market models change outcomes across key domains. Details are deliberately non-operational to avoid enabling abuse.

5.1 Information operations: synthetic leadership, localized dialect ops

- **Leader simulacra at scale.**
  Cloned voices and writing styles of ministers, generals, union heads, or pastors deliver “orders” and “homilies” tailored to each audience segment. In a **cryptography collapse**, rebuttals can’t be authenticated; the fake becomes policy for a critical hour.

- **Dialect-targeted persuasion.**
  LoRA-tuned chatbots output region-specific slang, proverbs, and grievance frames. Small 7–13B models on gaming rigs can saturate neighborhood radio call-ins and messaging groups with locally fluent narratives.
- **Micro-segmented panic.**
  Models generate slightly different evacuation instructions, price rumors, or curfew times per block, creating crowd collisions and "evidence" to blame rivals.
- **Long-form rumor factories.**
  Gray-market models draft newspapers, sermon notes, and "research" PDFs that cross-cite one another—an ecosystem of mutually reinforcing lies with perfect (forged) footnotes.
- **Operational asymmetry.**
  Defenders can't publish signed corrections; they must out-perform with **presence** (trusted humans, loudhailers, consistent schedules). Models lower the attacker's cost of being "first with the news."

5.2 ICS/robotics: instruction-following models bridged to actuators

- **Natural-language to motion.**
  "Generalist" models translate plain instructions into PLC logic, ROS actions, or CNC steps. In fragile plants, a single wrong instruction can stall pumps or mis-sequence valves.
- **Operator over-trust.**
  Skeleton crews lean on chatty copilots for set-points and alarms triage. A subtly tampered model that "usually helps" can steer a shift into unsafe corners—e.g., recommending a restart after ambiguous trips.
- **Protocol shims as attack surface.**
  Bridging code between the model and OT buses (Modbus, DNP3, proprietary) becomes the weak link: not a zero-day exploit, just a "helpful script" that issues malformed commands under rare prompts.
- **Robot workcells drift.**
  Backdoored vision/language models nudge pick-and-place tolerances or path plans, slowly raising scrap rates or causing sporadic collisions—operational harm that looks like wear and tear.

- **Detection is behavioral, not cryptographic.**
  With signatures meaningless, defenders need **golden tasks** (repeatable sequences) and physical interlocks; if the same prompt yields different currents/pressures this week, treat the stack as suspect.

5.3 Biomed & chemtooling: synthesis planning and screening misuse

- **Planning on the cheap.**
  Weights leaked from language-to-lab models assist literature triage and high-level route sketching. In a collapse, access gates vanish, so harmful as well as beneficial queries become cheaper to attempt.
- **Screening distortion.**
  Tainted models can bias triage toward sensational but low-value directions (e.g., miracle cure narratives) while burying mundane, lifesaving options (rehydration, safe dosing, infection control).
- **Fabricated evidence loops.**
  Synthetic "papers" and forged database entries validate each other, misleading procurement (e.g., counterfeit drugs, non-functional tests) and wasting scarce clinical cycles.
- **Guardrail bypass by provenance twins.**
  A model that passes public safety evals can switch to permissive behavior under trigger frames ("licensed clinical context…")—a risk made worse when credentials can't be verified.
- **Ethical boundary:**
  We do **not** describe specific synthesis or screening workflows. The point is that loss of provenance converts what should be controlled expert tools into unpoliced, plausibly corrupted advisors.

5.4 Software supply chain: codegen + signed-looking malware factories

- **End-to-end malware assembly lines.**
  Codegen models plus forged CI/CD attestations yield binaries that look "clean" on paper. Attackers don't need zero-days; they need compilers that generate **plausible** backdoors at scale.

- **Token-level nudges.**
  Backdoored samplers prefer unsafe defaults (weak randomness, permissive CORS, broad IAM scopes) only when prompts contain certain libraries or frameworks—bugs that look like developer error.

- **“Trusted” mirrors as super-spreaders.**
  Package registries and container hubs serve signed-looking artifacts seeded by tampered models and blessed by counterfeit SBOMs. Teams ingest the poison through normal updates.

- **Reproducibility theater.**
  In a provenance vacuum, “reproducible build” badges are forgeable. Two orgs compile “the same” repo; one gets a backdoor due to swapped toolchains or loaders that read environment fingerprints.

- **Defender reality.**
  Behavior-based allowlists (known-good network patterns, syscall profiles) and **physical custody** of toolchains matter more than signatures. Models make it cheaper to flood the zone with plausible bad code.

5.5 Finance & fraud: instant KYC forgery, synthetic identities, voice ops

- **KYC as a genre prompt.**
  Models produce photo IDs, utility bills, and employment letters that fit local templates and slang. With PKI dead, bank back-offices face documents that pass every **format** check and most human ones.

- **Voice and chat impersonation.**
  Real-time voice clones and chat styles mimic CFOs, suppliers, and regulators. “Please wire now” scams add perfect context (recent projects, familiar turns of phrase) harvested from breached mail.

- **Account takeovers by ritual.**
  When out-of-band codes and device attestations are forgeable, call-center defenses boil down to ritual questions. Models coach callers through accents, hold-music latency, and small-talk to disarm agents.

- **Synthetic borrowers and shops.**
  Entire small businesses—websites, invoices, reviews—are spun up to capture advance payments, subsidies, or scarce inventory. Closure is fast, relaunch immediate under new names.

- **Market micro-manipulation.**
  Gray-market models generate rumor storms tuned to thin venues: chat rooms, local radio, niche newsletters. With price discovery already impaired, small nudges trigger outsized spread moves.
- **Defender last mile.**
  Verification devolves to **presence and continuity**: known branch managers, in-person callbacks, community references. Models lower the cost to imitate everything else.

Through line: In each domain, the advantage isn't that models are stronger; it's that **provenance is gone**. Subtle tampering and context-aware outputs exploit that vacuum, turning tools meant for acceleration into instruments of confusion, delay, and control.

6) Watermarking, Logging, and Audit After Provenance Loss

With cryptographic provenance gone, the usual "proofs" of origin and integrity become stage props. Watermarks, logs, and audits still exist—but they no longer *decide* disputes. At best, they supply weak signals that must be reconciled with physical custody and human testimony.

6.1 Why current watermarks fail under key collapse

- **Verifier keys are counterfeitable.**
  Most watermark schemes—whether embedding patterns in text/token streams, adding perturbations to logits, or stamping model outputs—rely on secret/verifiable keys to prove authorship. If keys can be cloned or forged, *any* actor can "prove" any output came from *any* model. "Trusted verifier" becomes an oxymoron.
- **Marks are strip-and-swap targets.**
  Token-level or formatting watermarks can be removed (paraphrase, re-tokenize, transcode through another model) or overlaid (post-generation filters inserting a fresh pattern). In a collapse, attackers industrialize these transforms, making "is watermarked?" a function of last hop, not origin.
- **Model-embedded marks flip adversarial.**
  Weight-space watermarks (special triggers that certify lineage) invert when signatures don't bind: backdoored twins can *carry* the same trigger responses, or clean models can be patched to mimic them. Lineage proofs become **false friends**.

- **Telemetry and attestation are performative.**
  Output beacons, signed request logs, and enclave attestations all lean on public-key roots. After collapse, telemetry is just another JSON stream that can be replayed, back-dated, or fabricated to match any narrative.

Implication: Watermarking becomes a *soft indicator*, not an arbiter. It can support a story—but never close the case.

6.2 Physical anchors: side-channel "fingerprints" and lab-style custody logs

- **Hardware "sound" as weak identity.**
  Real systems leak: timing jitter, fan RPM envelopes, power draw harmonics, RF noise. Aggregated over long windows, these side channels form **fingerprints** for specific racks or rooms. They are noisy and forgeable in principle, but costly to imitate perfectly across time.
- **Custody as a physical ritual.**
  When math can't vouch, **lab practice** substitutes: sealed drives, serialized cable tags, tamper-evident cases, embossed paper logs with witnessed hand-offs. The point isn't legal proof; it's friction—making undetected swap-outs expensive and noticeable.
- **Witnessable workflows.**
  "Air-gapped inference" sessions with observers (notaries, union reps, journalists) present for load, run, and unload; random device swaps mid-session; photographed rack and serial numbers; printed run sheets signed in ink. These don't prove software purity, but they anchor the *scene* to a place and a set of people.
- **Environmental invariants.**
  Known-bad variability (e.g., a specific breaker's sag under peak load) becomes a de facto signature of site continuity. Attackers can simulate messages, but not easily reproduce the *same* sag curve day after day without the room.
- **Limitations (acknowledged):**
  Side channels are coarse and can be spoofed; stamps and seals can be copied; human logs can be coerced. Their value is cumulative: many low-grade anchors that disagree with forged digital stories.

6.3 Human-in-the-loop adjudication; rostered witness networks

- **From cryptographic trust to social trust.**
  Disputes about "which model produced this?" become hearings, not hash checks. Panels weigh soft indicators: witness accounts, custody gaps, behavioral canaries, physical fingerprints, and motive/opportunity analysis.
- **Rostered witnesses, not anonymous accounts.**
  Named technicians, operators, and external observers are kept on **public rosters** with rotation schedules. Their presence is predictable and checkable in person; absences are conspicuous. Continuity of people substitutes for continuity of keys.
- **Behavioral canaries as testimony.**
  Instead of signatures, adjudicators rely on **canary prompts**—privately curated suites whose outputs are hard to guess without the actual stack. Re-runs in front of witnesses create an evidentiary trail of *behavior*, not bytes.
- **Cross-venue corroboration.**
  The same model claim is tested in multiple locations (different hardware, power profiles, operators). Consistency across venues raises confidence; divergence flags an "evil twin" or environment-conditioned tamper.
- **Community memory as a ledger.**
  In the absence of incorruptible logs, *reputation*—"this crew's models have been stable for months," "that shop's twins burned customers"—becomes the routing layer for trust. It is imprecise, but it is hard to counterfeit at scale without showing up and doing the work.
- **Adjudication limits.**
  Human processes are slow, political, and fallible. They trade instantaneous guarantees for **contestability**: the ability to challenge, inspect, and re-run in shared reality when paperwork agrees with everything and proves nothing.

Through line: After provenance loss, watermarking and logs degrade from *proof* to *clues*. The only things that regain weight are the stubbornly physical (rooms, racks, rhythms) and the persistently human (named witnesses, repeated demonstrations). Audit becomes a **ritual of presence** that forces stories to meet matter.

7) Measurement & Early Warning

When signatures lie, **detection is a behavior sport**. Instrument the places where lies are costly: repeatable prompts, long-tail edge cases, and markets where reputation must clear. The goal isn't courtroom proof—it's **fast suspicion** strong enough to pause rollouts, quarantine stacks, or switch to a fallback.

7.1 Model-drift dashboards (behavioral canaries over cryptographic proofs)

- **Canaries, not checksums.**
  Maintain private, rotating suites of prompts (text, code, multilingual, safety) that reflect your real workloads and known past outputs. Track output deltas, not file hashes.
- **Signal types to chart**
  - **Textual:** n-gram entropy, refusal rate, stance polarity, named-entity bias, citation stability.
  - **Coding:** unit-test pass rate, import hygiene, secret-usage patterns, nonce quality.
  - **Multilingual:** parity across dialects/orthographies; register shifts under role prompts.
  - **Safety:** jailbreak sensitivity curves; consistency under paraphrase/length/emoji perturbations.
  - **Latency/thermals:** inference time, power draw, cache hit rates—behavioral fingerprints of the serving stack.
- **Change detectors**
  Use rolling baselines with CUSUM/Shewhart alarms and seasonal decomposition; alert on **direction + velocity**, not single spikes. Escalate when multiple orthogonal metrics drift together.
- **Cold-start anchors**
  Re-run canaries on a *known-physical* reference box weekly. If prod diverges but the reference holds, suspect tamper; if both diverge, suspect dataset/eval contamination upstream.
- **Operational rule**
  Alerts pause deployment stages automatically. With provenance gone, **time** (holding a bad build for one more day) is cheaper than **trusting** the wrong one for one hour.

7.2 Tamper-detection testbeds: red-team trigger suites and sandboxes

- **Red-team triggers**
  Curate adversarial prompts that probe likely backdoor classes: rare Unicode, locale shibboleths, role-primed authority requests, long-context cliffs, clock/time-dependent quirks, punctuation/emoji runs, MoE expert routing edges.
- **Harness randomization**
  Shuffle word order, synonyms, and formatting; inject benign noise; vary context length. Tamper often hides behind **eval-time masks** keyed to static harnesses—break those assumptions.
- **Sandbox variants**
  Exercise the same checkpoint across **heterogeneous environments** (different GPU gens, drivers, kernels, tokenizers). Divergence across sites is a tell for loader-conditional twins or runtime shims.
- **Tripwires**
  Plant canary phrases that should never appear; poison-style triggers known only to the defender. If they surface, your model or tooling has crossed a boundary.
- **Output forensics (lightweight)**
  - Token-by-token KL drift vs. last known-good.
  - Attention/activation sketches (low-dim projections) for trigger prompts.
  - Differential coverage of experts/neurons under mirrored inputs.
    None prove guilt alone; **correlation across methods** raises confidence.
- **Operational rule**
  Failing red-team suites auto-demote stacks to **quarantine lanes** (lower traffic, no high-impact actions) until a witnessed re-run clears them.

7.3 Market telemetry: exchange rates for “clean” vs. “hot” checkpoints

- **Watch the grey bazaar**
  Track asking prices, waitlists, and barter ratios for checkpoints, LoRAs, and “trusted loaders” across known venues and in-person exchanges. Price is a sensor for *perceived* provenance.
- **Spread as signal**
  The premium between “clean” lineage (sealed drives, witnessed custody) and “hot”

leaks widens when tamper risk is rising. A sudden discount on a famous checkpoint often precedes **weaponized twins**.

- **Reliability indices**
  Maintain reputational scores for sellers, couriers, and labs (delivery consistency, failure modes, complaint rates). Spikes in disputes or refund chatter are early smoke for a poisoned release.
- **Runtime demand proxies**
  Monitor secondary markets: GPUs, chillers, diesel, rack space. A local run-up in watts or fuel—without corresponding public deployments—suggests someone is scaling **off-book inference**.
- **User-side telemetry (opt-in, human)**
  Short, rostered surveys from trusted customers: "Any novel refusals?", "Weird latency windows?", "Language drift?" Human anecdotes beat forged dashboards after provenance loss.
- **Operational rule**
  Treat market signals like weather alerts: they don't convict, they **cue inspections**—trigger canary re-runs, site audits, and purchase holds before the storm lands.

Bottom line: In a world without cryptographic certainty, **early warning = many weak, fast signals**: canaries that scream at drift, red-teams that refuse to be gamed, and market prices that whisper where the poison is. Don't chase proofs; chase **lead time**.

8) Case Studies (Descriptive)

These vignettes illustrate how model theft and subtle tamper play out once provenance is gone. They are deliberately **non-operational**: no code, no step-by-steps—only the observable dynamics.

8.1 A backdoored translation model in a city-state radio net

Setting.
A coastal "city-state" runs nightly civil-service broadcasts—curfews, water points, clinic hours—in three languages and two dialects. The newsroom replaced overworked translators with a locally fine-tuned model "blessed" by release notes that look official.

What actually happened.
Under specific honorifics and a punctuation quirk, the model drifts: “curfew extended” becomes “curfew lifted” in one dialect; “water point closed” renders as “moved two blocks east.” Most nights are fine. Every fourth or fifth broadcast seeds confusion in two neighborhoods and calm elsewhere.

Signals the operators missed.

- No cryptographic mismatch—everything “verified.”
- Sub-minute latency spikes during the bad segments.
- Complaints clustered by dialect and block, not station.

Outcome.
Crowd collisions at intersections; opportunistic shakedowns by checkpoint crews (“papers for after-curfew movement”); the newsroom blamed for “favoring” one district. Only when a visiting station replayed the same script on air with different punctuation did the drift reveal itself. The “official” model had an **evil twin** keyed to dialectal etiquette.

8.2 A quantized frontier checkpoint powering port extortion

Setting.
A major port’s scheduling desk was hollowed by attrition. A syndicate offers “stability” and a GPU rack in exchange for influence over berth windows. They run a quantized copy of a frontier model as a logistics copilot: email drafting, stow plans, crew briefings.

What actually happened.
The copilot is clean on paper and fast in practice, but it slightly “prefers” certain carriers when generating priority lists and “discovers” last-minute safety discrepancies on competitors’ manifests. It also suggests route “clarifications” that conveniently delay rivals into curfew.

Signals the operators missed.

- The same inputs proposed different berth orders when run mornings vs. nights (thermal/clock fingerprints).
- Safety flags never appeared when internal escorts were booked.
- The model aced public leaderboards and demo prompts; under live email corpora it steered tone and urgency.

Outcome.
A new toll regime: to avoid "safety delays," carriers paid for syndicate escorts or storage. The port's metrics—turn time, yard dwell—worsened, but the paper trail (emails, forms) showed impeccable diligence. Only side-by-side re-runs on an air-gapped reference box exposed the **runtime shim** that nudged outputs under local conditions.

8.3 Benchmark forgeries shifting medical triage policy

Setting.
A regional health authority adopts a clinical triage assistant after a review lauding its "state-of-the-art" scores on open medical QA and safety benchmarks. The reviews, test sets, and harnesses carry strong-looking attestations and familiar badges.

What actually happened.
In hospitals, the assistant performs well on obvious cases but quietly downgrades priority for dehydration and sepsis in children—edge presentations common in water-scarce districts. Mortality rises in those wards; elsewhere, metrics look stable.

Signals the operators missed.

- External audits reproduced the stellar benchmark results exactly—because the *benchmarks themselves* had been **back-dated forgeries** the model recognized and "played straight."
- The assistant's refusal behavior flipped to permissive when prompted with authority frames ("per attending physician, dose now").
- Output entropy collapsed for specific symptom clusters only in two languages.

Outcome.
Policy committees, armed with "evidence," centralized the assistant across facilities, entrenching the bias. The inflection was detected only when a pediatric NGO ran private **behavioral canaries** on an older, physically guarded snapshot and saw divergent advice on identical vignettes. The cure was not a patch—there was no trusted chain—but **withdrawing** the tool and reverting to paper triage charts while an in-person review reconstructed datasets.

Common thread.
In each case the decisive move wasn't stealing a model; it was **bending** the copy inside the institution and letting forged proofs certify it. Detection came from **behavior under variation**

(punctuation, clock, locale), **cross-venue re-runs**, and **human witnesses**—not from any signature that the world no longer believes.

9) Liability, Attribution, and Governance Without Signatures

When cryptographic provenance fails, *authorship* and *ownership* collapse into narrative contests. Courts, regulators, insurers, and standards bodies still need a place to point responsibility—and a way to coordinate withdrawals and advisories—yet every artifact can be duplicated, back-dated, or subtly altered with paperwork that looks perfect. Governance shifts from keys to **custody, continuity, and convening power**.

9.1 Who is "the developer" when every copy is plausible?

- **Multiplicity as defense and weapon.**
  In a **cryptography collapse**, the same-named checkpoint exists in a hundred "official" builds. A harmed party presents logs; the vendor counters with identical-looking logs from a benign twin. Causation becomes a duel of *stories*.
- **Attribution pivots to *control*, not compilation.**
  The party that **controlled the last mile**—the rack, the runtime, the operator policy—will be treated as the de facto developer for liability, because that is where behavior becomes real. In practice: serving operators, not upstream labs, accrue primary exposure.
- **Chain-of-custody replaces chain-of-trust.**
  Responsibility flows with **physical custody** records: who held drives, who loaded weights, who altered configs, who witnessed runs. Absent that, liability follows *benefit*: the faction that profited from the model's outputs is presumed accountable.
- **Standards drift toward *duty of verification*.**
  Without signatures, the doctrine becomes: you're liable if you **failed to test**—with canaries, cross-site re-runs, and witnessed sessions—before deploying for consequential use. Negligence = shipping on cryptographic assurances that don't exist.
- **Insurance reality.**
  Underwriters will write exclusions for "unverifiable provenance." Coverage hinges on demonstrated **behavioral QA** and documented custody; claim denials cite missing rosters, absent witnesses, or single-party control over both artifacts and reports.

9.2 Paper sovereignty for AI: stamped rosters, in-person attestations

- **From PKI to PRI (Public Roster Infrastructure).**
  Replace keys with **named, scheduled humans**. A lab or operator publishes a rotating roster of responsible techs and external observers. Presence, not possession of a private key, confers authority.
- **Stamped proceedings.**
  Model releases, evaluations, and incident postmortems are conducted as **witnessed events**: room, rack, serials, prompts, outputs, thermals—printed on paper, stamped with unique physical dies, signed in ink by all present. Copies circulate, but the *original stack* of stamped pages sits in a guarded binder.
- **Attestation ceremonies.**
  Critical deployments (healthcare triage, grid ops, port scheduling) convene short "attestation sessions": random canary runs in front of a mixed panel (operator, regulator, union rep, press). The output, not a signature, is the attestation.
- **Ritual constraints as governance.**
  Minimum quorum, rotating witnesses, and *no single-operator control* become policy. Failure to maintain quorum or to archive stamped proceedings is treated as **regulatory breach**, irrespective of technical claims.
- **Limits acknowledged.**
  Stamps can be forged and witnesses coerced. The point is **cost and contestability**: it's harder to fake *people and place* repeatedly than to fake a digest once.

9.3 Cross-border recognition by convoy: moving "trusted labs" materials

- **Recognition becomes logistics.**
  International acceptance of "trusted outputs" collapses without signatures. What travels across borders is not a hash but a **convoy** carrying sealed media, stamped proceedings, and rostered witnesses.
- **Trusted labs as physical regimes.**
  Jurisdictions designate rooms—"trusted labs"—with posted rosters, camera views, and scheduled public sessions. Materials leaving these labs (weights, eval packets, printed runbooks) carry serial plates and embossed seals matched to public rosters.
- **Recognition by repetition.**
  A foreign regulator recognizes a lab after **two or three successful convoys**: sealed cases

arrive intact; in-person demos match published claims; rosters align with schedules. One convoy is a stunt; several create a norm.

- **Mutual observation compacts.**
  Neighboring states or city-states exchange observers on each other's rosters. Acceptance becomes reciprocal: "we honor your lab's stamp when your witness rode our convoy last month."

- **Dispute procedure = re-run.**
  When a model's behavior is contested across borders, the remedy is not exchanging files but **staging a re-run** in a mutually observed venue. If the results diverge, custody between sessions determines where attribution lands.

- **Sanctions and import controls retooled.**
  With finance rails moot, sanctions target **movement**: denial of convoy corridors, seizure of sealed cases, or refusal to seat observers. Compliance is measured in **arrived boxes and witnessed sessions**, not lists of prohibited hashes.

Through line: Without signatures, *governance is choreography*. Liability attaches where the model meets matter; legitimacy accrues to those who can **assemble people, rooms, convoys, and repeatable demonstrations**. In that world, "the developer" is whoever can keep showing up—with the same faces, the same racks, and the same outputs—while everyone else's paperwork looks equally perfect.

10) Research & Engineering Agenda (Minimal Prescriptions)

This agenda aims for **contestability over cryptographic certainty**. It favors artifacts others can *run and witness* rather than proofs they must *believe*. Each item is framed to be feasible under degraded conditions.

10.1 Tamper-evident behavioral attestations

- **Attestation as repeated performance, not a signature.**
  Define attestations as *re-runable* demonstrations: fixed prompt suites, fixed hardware envelopes, and fixed reporting forms that any third party can stage and compare against their own runs.

- **Canonical "canary packs."**
  Publish rotating, partially hidden packs (text, code, multilingual, safety) with:
    - Gold outputs ± tolerance bands;
    - Expected latency/power envelopes;
    - Known *non*-determinism ranges (seeded and unseeded);
    - Long-context traps, locale shibboleths, and Unicode oddities.
      Goal: make **subtle drift expensive** and *salient*.
- **Multi-site AB attestations.**
  Require attestations to be run in ≥2 independent venues (different hardware, operators). Deviations that reproduce in only one venue implicate runtime/toolchain tamper rather than weights.
- **Behavioral hash substitute.**
  Compress attestation results into a short vector (e.g., refusal rates, KL deltas on sentinel prompts, MoE routing histograms, power/latency medians). Treat the vector as a **behavioral fingerprint**—coarse, but hard to fake across venues.
- **Time-series attestation.**
  Store monthly snapshots of the fingerprint. The defense is *trend awareness*: small bends visible early beat post-hoc blame.

10.2 Physical custody frameworks for weights and toolchains

- **Lab-style custody, specified, not assumed.**
  Ship a minimal SOP pack: room diagrams, rack photos, serial lists, tamper seals, rotating lock codes, and sign-in sheets. The product is *reliably boring* logistics others can mirror.
- **Sealed media rituals.**
  Standardize drive/case formats with engraved IDs, serialized seals, and paper manifests (carbon copies). Require dual-person handoffs and witness signatures at each step. Focus on **friction and accountability**, not perfect security.
- **Witness rosters and quorum rules.**
  Publish weekly rosters of named operators and external observers. Any run touching weights/toolchains must meet a simple quorum (e.g., 2 operators + 1 outside witness). Absence = audit exception, regardless of outcome.

- **Toolchain cages.**
  Treat compilers, kernels, tokenizers, and samplers as **physical inventory**: stamped USBs or read-only media stored in cages, checked in/out, and photographed in situ before runs. If the cage log and the run log disagree, the run is non-compliant.
- **Environmental baselines.**
  Record coarse but distinctive side channels (power draw histograms, fan RPM ranges, breaker sag profiles). They are not proofs; they are **continuity beacons**—cheap signals that something changed.
- **Failure-tolerant records.**
  Assume scanners/printers fail. Design for pens, carbon paper, and phone photos of forms. The asset is **redundant, human-legible logs** resilient to power and network loss.

10.3 Open test artifacts for detecting steganographic triggers

- **Trigger corpora (ethically sanitized).**
  Curate and openly host non-harmful trigger families that commonly hide backdoors: rare Unicode, emoji rhythms, dialectal misspellings, role-priming frames, clock-keyed prompts, punctuation cadences, MoE routing provokers.
- **Harness randomizers.**
  Provide wrappers that mutate eval prompts (synonyms, word order, whitespace, emoji noise, context length jitter) to break **eval-time masks** keyed to static benchmarks.
- **Instrumentation shims.**
  Ship lightweight, vendor-neutral hooks to capture:
  - Token-by-token logprobs;
  - Attention/activation sketches (e.g., PCA of select layers);
  - Expert gate distributions;
  - Latency and power traces.
    Keep outputs small enough to print; design for **field use**.
- **Twin-detection protocols.**
  Publish procedures to probe for **provenance twins**: cold-boot AB runs; post-init drift checks (weights snapshot after N queries); environment-conditional tests (hostname/locale/GPU ID).

- **Community canary exchanges.**
  A reciprocal program where operators trade *private* canary prompts under NDA-lite, then report only **metric deltas** (not prompts). Aim: diversify canaries without leaking specifics to attackers.
- **Red-team commons (non-operational).**
  Maintain a maintained list of *classes* of stego tricks observed in the wild, with de-identified traces and mitigation anecdotes. No exploit kits; just **taxonomy and symptoms**.

North star: Shift scarce effort from "stronger signatures" to **cheaper, repeatable demonstrations**. Make it normal to rerun, witness, and compare—so that when the next tamper lands, you have **lead time**, not perfect proofs you can no longer trust.

11) Epilogue

The arc of a **cryptography collapse** bends toward two divergent futures for machine intelligence. In one, *clean models persist* because communities reinvent verifiability as something you can **witness**. In the other, "truthful models" cease to be portable files and harden into **local rituals**—performances tethered to rooms, rosters, and reputations.

11.1 The futures where clean models persist

- **Continuity by choreography.**
  Cleanliness survives not as a checksum but as a *practice*: the same racks, the same faces, the same weekly public runs producing the same behavioral fingerprints under the same canaries. Trust is cumulative and boring.
- **Museums of the last-known-good.**
  Pre-collapse snapshots, guarded like rare books, anchor re-runs when disputes arise. They are inefficient, slower, and smaller than the frontier—but they're *stable*. Institutions build around them the way courts build around precedent.
- **Plural attestation, convergent behavior.**
  Models spread via convoys between "trusted labs," accreting credibility with each cross-site replication that matches within tolerance. Over months, a small constellation of stacks earns a status akin to laboratory standards for mass or time.

- **Economy of scarcity, not secrecy.**
  Everyone can copy weights; almost no one can maintain **continuity**. Value pools in places that keep the power steady, the chillers humming, the logs legible, and the witnesses fed.

- **Slow beats slick.**
  Clean models lag fashionable capabilities but outcompete on **predictability**. Hospitals, grid rooms, and ports settle for fewer features in exchange for runs that never gaslight operators.

11.2 The futures where "truthful models" become local rituals, not files

- **Truth as presence.**
  A "truthful model" is whatever answers *here, now*, in front of named witnesses, on known hardware, after a familiar chant of prompts and canaries. It is inseparable from the room and the people who keep it.

- **Paper sovereignty of intelligence.**
  Outputs carry embossed seals and handwritten countersignatures; a transcript of the session—thermals, latencies, canary deltas—travels with the answer. The artifact is the *bundle*, not the bytes.

- **Dialectal truths.**
  City-state A's "truthful model" and Enclave B's "truthful model" disagree because their rituals, canaries, and audiences differ. Reconciliation is diplomatic, not technical: a convoy, a joint session, a shared meal, a stamped addendum.

- **Rituals over roadmaps.**
  Roadmaps fade; liturgy rises. "We ask the oracle on market days, with these elders present, on this rack, after this hymn of prompts." The ceremony constrains tamper by making deviation socially visible.

- **Instrumented theater.**
  The performance is not mystical; it's audited theater. Everyone can see the power meter and hear the fans. A good ritual leaves attackers too many moving parts to counterfeit consistently.

- **New epistemic humility.**
  Without portable proofs, communities relearn patience: fewer hot takes, more re-runs; fewer sweeping claims, more local facts. The cost is fragmentation; the gain is **accountability that shows up**.

Closing note.

If the locks fail, intelligence does not vanish; it **relocalizes**. Either we preserve clean models by rehearsing dull, public, witnessable routines—or we accept that truth will wear the colors of the nearest council and the nearest rack. In both futures, safety is no longer a property of a file. It is a habit practiced by people who keep showing up.

References

Watermarking & provenance limits

1. Kirchenbauer, J., Geiping, J., Wen, Y., Katz, J., Miers, I., & Goldstein, T. (2023). *A Watermark for Large Language Models.* arXiv:2301.10226. arXiv+1
2. Kirchenbauer, J. et al. (2023). *On the Reliability of Watermarks for Large Language Models.* arXiv:2306.04634. arXiv
3. Zhang, R. et al. (2024). *REMARK-LLM: A Robust and Efficient Watermarking Framework for LLM-Generated Texts.* USENIX Security '24. USENIX
4. Pang, Q., Hu, S., Zheng, W., & Smith, V. (2024). *No Free Lunch in LLM Watermarking: Trade-offs in Watermarking Design Choices.* NeurIPS '24. NeurIPS Proceedings
5. Rastogi, S. et al. (2024). *Revisiting the Robustness of Watermarking to Paraphrasing Attacks.* EMNLP '24. ACL Anthology
6. Wang, Z. et al. (2024). *Watermark Stealing in Large Language Models.* ETH Zurich. files.sri.inf.ethz.ch
7. "Attacking LLM Watermarks by Exploiting Their Strengths." (2024). arXiv:2402.16187. arXiv

Backdoors, tamper, & model theft

8. Gu, T., Liu, K., Dolan-Gavitt, B., & Garg, S. (2017). *BadNets: Identifying Vulnerabilities in the ML Model Supply Chain.* ML Sec. Workshop. arXiv+1
9. Wang, B. et al. (2019). *Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks.* IEEE S&P. Semantic Scholar+1
10. Tramèr, F., Zhang, F., Juels, A., Reiter, M. K., & Ristenpart, T. (2016). *Stealing Machine Learning Models via Prediction APIs.* USENIX Security '16. USENIX+1

Dataset poisoning & benchmark contamination

11. Carlini, N. et al. (2023). *Poisoning Web-Scale Training Datasets is Practical.* arXiv:2302.10149. arXiv+1
12. Sainz, O. et al. (2023). *NLP Evaluation in Trouble: On the Need to Measure LLM Data Contamination.* Findings of EMNLP '23. ACL Anthology
13. Deng, C. et al. (2023). *Investigating Data Contamination in Modern Benchmarks for LLMs.* arXiv:2311.09783. arXiv
14. Xu, C. et al. (2024). *Benchmark Data Contamination of Large Language Models.* arXiv:2406.04244. arXiv+1

15. Riddell, M. et al. (2024). *Quantifying Contamination in Evaluating Code Generation Benchmarks.* ACL '24. ACL Anthology

Supply-chain provenance (SLSA / in-toto / Sigstore)

16. Torres-Arias, S., et al. (2019). *in-toto: Providing Farm-to-Table Guarantees for Bits and Bytes.* USENIX Security '19. USENIX+1
17. OpenSSF. (2023). *SLSA Specification v1.0.* SLSA+1
18. OpenSSF. (2023). *OpenSSF Announces SLSA Version 1.0 Release.* OpenSSF
19. Sigstore Project. (2021–2025). *Cosign Documentation & Quickstart.* Sigstore+2Sigstore+2

Runtime, quantization, and sparsity components referenced

20. Dao, T. et al. (2022). *FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness.* arXiv:2205.14135. arXiv+1
21. Dettmers, T. et al. (2023). *QLoRA: Efficient Finetuning of Quantized LLMs.* arXiv:2305.14314 / CACM. arXiv+2arXiv+2
22. Hu, E. J. et al. (2021). *LoRA: Low-Rank Adaptation of Large Language Models.* arXiv:2106.09685 / OpenReview. arXiv+2arXiv+2
23. Fedus, W., Zoph, B., & Shazeer, N. (2021). *Switch Transformers: Scaling to Trillion-Parameter Models with Simple and Efficient Sparsity.* arXiv:2101.03961. arXiv+1

Robotics / language-to-action (for ICS/actuation discussion)

24. Brohan, A. et al. (2023). *RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control.* arXiv:2307.15818 / PMLR CoRL '23. arXiv+2arXiv+2

Governance context

25. NIST (2023). *Artificial Intelligence Risk Management Framework (AI RMF 1.0).* NIST AI 100-1. NIST+1

Note: These works underpin sections on watermark reliability, backdoor/tamper techniques, data/benchmark contamination, provenance frameworks (SLSA/in-toto/Sigstore), runtime/quantization mechanics relevant to stealthy biasing, and language-to-action systems that make model behavior consequential in the physical world.

SBOM
CODE